CS224 Assignment 2

180070054 *(Shreya Laddha)* | 170100035 *(Tezan Sahu)*

Task 1:

1. The trace has all sorts of packets but is overwhelmingly made of a trace of a connection related to one particular activity or command. What is that?
   HTTP GET command

2. What is the IP address, port number of the two ends of this connection? How many packets do you see of this connection (hint: filter properly and see bottom right of wireshark window). Let us call this Connection X. The following questions are about this Connection X.
   a. At which IP address and port number is the "server" running? What is the application layer protocol in connection X?
   b. At which IP address and port number is the client running? What possible terminal command at the client could generate such a trace? Write the whole command.
   c. Can you see the client hostname in the trace? If yes what is it? The server hostname? If yes, what is it?

   192.168.0.106 (port 50718) and 128.171.224.109 (port 80) are the two ends of this connection. Total packets = 948987 in this connection.
   a. *Server is running at 128.171.224.109 (port 80). HTTP is the application layer protocol.*
   b. *Client is running at 192.168.0.106 (port 50718). `wget` and `curl` command can be used to send HTTP GET request in Linux. The given packets are generated by a Wget application user agent, version 1.17.*
      `wget` http://www.honolulu.hawaii.edu/sites/www2.honolulu.hawaii.edu/files/policies-nondiscrimination.pdf
   c. *Client hostname - Sandips-iMac-3.local*
      Server hostname - www00.honolulu.hawaii.edu

3. Look at packets 5 and 6. What question is Packet 5 asking?**(0.5)** What answer did it get in Packet 6?
   Question asked in Packet 5 - Who has 192.168.0.1? Tell 192.168.0.106 (it is asking for the MAC address of the host with IP address = 192.168.0.1)
   Answer in Packet 6 - 192.168.0.1 is at 48:ee:0c:46:ab:2c

4. Look at packets 7,8,9,10. Describe what they are doing and which is the client, which is the server, and what is this kind of server called.

The packets are a part of DNS query-response system, client asking for the IPv4 and IPv6 Address related with a fully qualified domain name (have to frame it better later).
***Packet 7** - Standard DNS Query asking for IPv4 address of www.honolulu.hawaii.edu*
***Packet 9** - Response of Packet 7 query*
***Packet 8** - Standard DNS Query asking for IPv6 address of www.honolulu.hawaii.edu*
***Packet 10** - Response of Packet 8 query*
***Client** - 192.168.0.106*
***Server** - 192.168.0.1*
*This server is called **DNS nameserver**.*

5. For packet number 14 (which is of Connection X), answer the following questions.
    a. The headers of which layers can you see clearly? Write 3 header field names and their values for each of the layers that you can see in this packet. E.g. For each layer, you can answer in the following format:

    Application Layer (HTTP):
    Host = www.honolulu.hawaii.edu\r\n
    Accept-Encoding = identity\r\n
    *Accept = */*\r\n*

    Transport Layer (TCP):
    Source Port = 50718
    Destination Port = 80
    Window = 229

    Network Layer (IP):
    Total Length = 269
    Time to live = 64
    Header checksum = 0x00008498

    Link Layer (Ethernet):
    Type = 0x00000800
    Source = cc:3d:82:9d:a9:cc
    Destination = 48:ee:0c:46:ab:2c

    b. The packet number 14 is going from which host (IP addr) to which host (IP addr)?
    192.168.0.106 (source) to 128.171.224.109 (destination)

    c. The packet number 14 is going from which network interface (MAC address) to which network interface (MAC addr)?
    cc:3d:82:9d:a9:cc (source) to 48:ee:0c:46:ab:2c (destination)

6. Using information in packet numbers 5,6 and 14 (and answers to above questions) can you infer whether there is a direct link between the two end-hosts of Connection X. Justify your answer. Do not use any reasoning other than what is evident in these packets.

   The two end hosts of the connection are not connected by a direct link.
   Reason: *The destination MAC address in Packet 14 is the same as the MAC address of the DNS Nameserver (D-LinkIn_46:ab:2c), which was obtained from packets 5 & 6. Hence, this is the "next hop" from the client to the server. Had there been a direct link, the destination MAC address would have been that of the server (128.171.224.109).*

   a. If there is no direct link, what is the IP address and MAC address of the next hop from the client?
      IP address - 192.168.0.1
      MAC address - D-LinkIn_46:ab:2c (48:ee:0c:46:ab:2c)

The following questions need you to look at timestamps and do some calculations. You are advised to use a spreadsheet (e.g. Libreoffice calc) for all these otherwise the calculations will take time, and will have to be repeated. Also, it might be a good idea to filter the trace on the server IP address.

7. Find packet pairs in the setup and teardown of Connection X that represent roundtrips from the client to server to client. Find 3 such packet pairs. Write the packet number pairs.

   Setup: Packets 11 and 12
   Teardown: Packets 970 & 971, Packets 971 & 972

   a.
   b. Calculate the RTT (roundtrip) you are getting from each of these 3 (write which packet pair gives which time). Write the min, max and average.
      11 & 12: *409.21 milliseconds*
      970 & 971: *407.17 milliseconds*
      971 & 972: *32 microseconds*

      Min RTT *= 326 microseconds*
      Max RTT *= 409.2115 millimicroseconds*
      Average RTT *= 272.1412 millimicroseconds*

Now look at packets from the server to client in packet number range 18 to 32.

8. How many packets came **to** the client **from** the server in this range?
   8 packets in total came to the client from server, i.e every alternate packet was from server to the client (considering inclusive range 18 to 32)

9. What is the interarrival time of packets coming **to** the client **from** the server? Write all the times, the average, min and max.

Packets	***Arrival time [s]***	***Interarrival time [microseconds]***
Packet 18	*3.009889*	
Packet 20	*3.009918*	*29*
Packet 22	*3.018752*	*8834*
Packet 24	*3.018935*	*183*
Packet 26	*3.021617*	*2682*
Packet 28	*3.021651*	*34*
Packet 30	*3.023456*	*1805*
Packet 32	*3.023483*	*27*

***Average** = 1942 microseconds (0.001942 s)*
***Min** = 27 microseconds (0.000027 s)*
***Max** = 8834 microseconds (0.008834 s)*

Now observe packets 32 onwards

10. Packet numbers 32 and 34 are sequential arrivals to the client. What is their interarrival time?
    Interarrival Time: 0.428091 seconds

    a. What inference can you make from this and the interarrival time of packets 18 to 32? Specifically, do you think the server is using stop and wait protocol to send its data? (You may assume that RTT from server-client-server will have a similar value to client-server-client RTT). If not, what might be the window size (in units of packets). Justify your inference.
    The average interarrival time of packets 18-32 (0.001942 s) is much smaller compared to the interarrival time between packets 32 & 34 (0.428091 s).
    This indicates that the server uses a sliding window (& not a stop & wait protocol) for transmitting data to the client. The server does not wait for the ACKs of the client until it sends data corresponding to its window size (packets 18, 20, 22, …, 32). Finally, when it receives the ACK for packet 18 (after ~0.4 s, which is close to the average RTT), it slides its window & sends new packets.
    The window size is 8 (packets 18, 20, 22, …, 32)

b. How much data in bytes do you think the server sent without waiting for an acknowledgement? (Hint: for this you can use the TCP sequence number seen in the packets. TCP numbers its packet sequence numbers in units of bytes. E.g. The first data packet has sequence number 1. Now, if Packet P_k has Sequence number S_k (bytes) and size L bytes, the next packet P(k+1) will have Sequence number S_k +L bytes.
The server sent packets 18, 20, 22, …, 32 without waiting for an acknowledgement.
The relative sequence number of Packet 18 is 1 (starts at 1 byte).
The relative sequence number of Packet 32 is 11395 (starts at 11395 bytes)
The length of Packet 32 is 1448 bytes
Thus, the total data sent without waiting for acknowledgement = (11395 + 1448) - 1 = 12842 bytes

11. Find series of packets with the same arrival pattern as packets 18 to 34. Find three more such series. Using TCP sequence numbers, figure out how much data the sender is sending (essentially its window size in bytes) without waiting for an ack. Fill in the following table:

Packet series start packet number	***Series end packet number***	***Window size in bytes***
34	*52*	*24120*
52	*64*	*37648*
64	*76*	*40544*

What inference can you draw from this table about the window size (in bytes) the sender is using?
Window size is progressively increasing (could be attributed to the increase in congestion window size of TCP)

12. What is the “raw” throughput achieved in this connection? (Raw throughput can be calculated e.g. by bytes sent in packets 18-32 and time in which they were received).
Total bytes sent: *12842*
Time: *0.013594 seconds*
Raw Throughput *= 944681.477 KBps = 7.557 Mbps*

13. What is the latency from the connection setup request by the client (“SYN”), to getting the first packet of the file?
0.785 s

14. What is the latency from the connection setup request by the client (“SYN”), to getting the last packet of the file?
    14.330 s

15. What is the effective throughput of the whole connection?
    [Data received / (Receiving time of Last packet of file - connection setup request time)]
    2557890 bytes / 14.330 s = 178.499 KBps = 1.428 Mbps

16. What do you think is dominating the latency? (slow data rate of some bottleneck link? Or large Round Trip Time)
    Observation: *Time from client to server ~ 0.4 s, while time from server to client ~ order of microseconds*
    This may indicate that the large RTT which dominates the latency can be due to a bottleneck link lying in the path from the client to the server. Although the packets from the server to the client do not traverse this bottleneck link (perhaps due to different entries in the forwarding table of the router for the reverse path).

Task 2:

1. What is the MAC address of my laptop?
   00:22:fa:2b:4f:50

2. What is the MAC address of the wireless interface of WiFi access point?
   94:d7:23:7b:e1:90

3. What do the first four packets of the laptopConnectingToWireless.pcap trace seem to be doing?
   4-Way EAPoL Handshake to generate some encryption keys which can be used to encrypt actual data sent over Wireless medium

4. What is the subnet address of my home wireless network (write in a.b.c.d/x notation)?
   192.168.0.0/24

5. My laptop is getting an IP address through DHCP. How many packets are involved in getting this address? Why does it seem to be less than the expected number of packets?
   Number of packets involved: *2 (Packets 5 & 6)*
   These packets correspond to the DHCP Request & Ack messages. The initial DHCP Discover & Offer messages are missing here (add a probable reason for this). Hence it seems lesser than the expected number of packets.

6. What is my laptop's IP address?
   192.168.0.2

7. What is the IP address of the wireless interface of the Wifi access point?
   192.168.0.1

8. The access point (router) is connected to the WAN, i.e.. the "rest of the Internet" through a wired interface
    1. What is the IP address of this interface?
       180.151.244.118

    2. What is the subnet number of the network that this interface connects to? (in a.b.c.d/x notation)
       180.151.244.0/23

2. What is the IP address of my laptop as seen on "login.iitb.ac.in"?
   180.151.244.118 (Can be confirmed using lines 81 & 82 in netstat_surya.out)

10. Where do you think the NAT is happening for my laptop's address? Explain why you think so.
    At the WiFi Router (IP: 192.168.0.1)
    Reason: *192.168.0.1 is the default gateway to the internet for the home wifi network. Also, the IP address of the laptop, as seen on login.iitb.ac.in corresponds to the IP Address of the interface of the wifi router facing the external internet.*

11. What is the actual port number on the laptop of the first ssh connection?
    1158

12. What is the actual port number on the laptop of the second ssh connection?
    1159 [according to laptopSSHtoLoginIITB.pcap]

13. What must be the NAT table entry (in my ISP's NAT) corresponding to the ssh connections from my laptop? (Show both entries)

Private Network Side	***Global Internet Side***
192.168.0.2: 1158	*180.151.244.118: 1060*
192.168.0.2: 1159	*180.151.244.118: 1077*

14. What is the network address of the network that login.iitb is on? (a.b.c.d./x notation)
    10.105.1.0/24

    1. What is its default router address?
       10.105.1.1

    2. What is the default router's MAC address?
       3Com_03:01:69 (00:01:02:03:01:69)

15. What is login.iitb.ac.in's internal (private) IP address? What is the actual port number at which the ssh connection is made?
    Internal IP Address: *10.105.1.14*
    Actual Port for SSH Connection: *22*

16. What is login.iitb's IP address & port number for the two ssh connections - AS SEEN BY my laptop?
    IP Address: *103.21.126.139,* ***Port Number:*** *5022 (For both SSH connections)*

17. What do you think is the table entry in IITB's NAT corresponding to the two ssh connections from my laptop to login.iitb.ac.in?
    IITB's NAT table contains just 1 entry for the port running SSH service. This seems to be shared by both the SSH connections made from "my" laptop:

Private Network Side	***Global Internet Side***
10.105.1.14: 22	*103.21.126.139: 5022*

18. Label the diagram on the next page with all the interface addresses that are possible to be known (write the actual addresses, not NAT translations). Some samples of how to show this are shown, you have to show this for all interfaces that can be labeled with the given data. Also, write the approximate RTTs between the hops in the space shown. You

may use multiple sources of information for this purpose – which may not have been taken for the same time.

Hop Count	***Sender IP***	***Receiver IP***	***Hop RTT***
1	*192.168.0.2 (Laptop)*	*192.168.0.1 (Gateway)*	*1 ms*
2	*192.168.0.1 (Gateway)*	*180.151.244.1*	*8.33 ms*
3	*180.151.244.1*	*125.63.82.209*	*1.67 ms*
4	*125.63.82.209*	*203.122.61.73*	*29.33 ms*
5	*203.122.61.73*	*218.100.48.28*	*(Can't calculate)*
6	*218.100.48.28*	*182.19.105.73*	*14.33 ms*
7	*182.19.105.73*	*115.114.89.38*	*(Can't calculate)*
8	*115.114.89.38*	*...*	*NA*
NA	*...*	*10.201.250.100*	*NA*
NA	*10.201.250.100*	*192.0.20.2*	*(Can't calculate)*
NA	*192.0.20.2*	*10.250.105.1*	*0.759 ms*
NA	*10.250.105.1*	*10.105.250.1*	*(Can't calculate)*
NA	*10.105.250.1*	*10.105.1.14 (Private IP of login.iitb.ac.in)*	*0.828 ms*

Note: *We mention "Can't calculate" when the average RTT for hop N is lesser than the average RTT for hop N-1.*

1. Find one (or more) samples of RTT between my laptop and login.iitb.
   39.56 ms (based on iRTT between hosts as seen in packet 453)

2. State whether you think the total RTT correlated well with the individual hop RTTs, if not why?
   The total RTT obtained from the sum of RTTs of the hops (> 56.247 ms) is greater than the RTT obtained using the packet trace (39.56 ms).

3. Suppose you are forced to guess the number of hops in the "unknown" area – how many do you think there are, and what is your reasoning behind your guess?
   -

Task 3:

1) Compare the sshToSurya trace with the scpToSurya trace. There is a difference in TCP flag settings in these traces. What is it and why is it so?
Observation: *The scpToSurya trace mostly has only the ACK flag set, while the sshToSurya trace has the ACK & the PSH flags set in almost every packet sent from the client to surya.*
Reasoning: *SCP involves file transfer & the data sent to surya need not be sent to the application immediately. The application can read data from the buffer slowly thereafter. But, in SSH, commands being typed from the client must appear on the terminal as soon as possible. Thus, the data needs to be sent to the application at the earliest. Thus, PSH flag being set for SSH connection is justified. Also, SSH commands may not be large enough to fill up the MSS, yet, they must be sent ASAP, unlike a file in SCP.*

2) In the scpToSurya trace.

1. What is the point at which you can be sure that the TCP congestion window size has increased beyond 1 packet? (State wireshark packet number range, which represents the "window full of packets"). At this point, what could be, in terms of number of packets, the window size?
   -

2. What is the smallest frame length? Briefly explain it.
   54 bytes
   These are ACKs to Data without any extra data piggybacked (It contains only headers of different layers)

3. What is the maximum frame length? Briefly explain it.
   1514 bytes
   Corresponds to MSS of 1460 (with headers added), as advertised in the Options while setting up the TCP connection.

4. At which packet number do you think the actual file transfer started (as opposed to tcp and scp setup messages)? Explain why.
   Packet Number 71
   Reason: *This is the first of the series of packets with size equal to the MSS (although there are occasional instances after that also where length < MSS), indicating that the laptop has sufficient data (ie, the file) to fill a TCP segment & SCP it to surya. Moreover, it is from Packet 71 that we do NOT see the regular PSH flag set (setting of the PSH flag would indicate that the received data should be passed to the receiving application immediately, which would be necessary for setup commands, but not for file transfer).*

3) Now look at the scpFromSurya set of traces. (suryaTrace, and laptop trace).

3.1) In the suryaTrace, look at packet number 1590. It has been tagged by wireshark as a Duplicate Ack.

3.1.1) This indicates the delay/loss of which packet? (Refer to packets by their TCP sequence numbers). (1 mark)
This indicates the delay/loss of packet number 1523 (TCP relative sequence number = 69978)
Reason: *Packets till packet number 1521 (TCP relative sequence number = 67058) have been ACKed.*

3.1.2) What else does an arrival of a duplicate ACK indicate?
It indicates ***congestion in the network****, leading to potential dropping of packets at buffers of intermediate routers (not at end host, since the advertised window size is still large enough).*

3.1.3) At this point, how many bytes have been sent by surya but not acknowledged yet?
Packets till 1521 (relative sequence number = 67058) have been ACKed, while packets till 1589 (relative sequence number = 113778, length = 2920) have been sent from the client.
Thus, the total unACKed data = (113778 + 2920) - 67058 = ***49640 bytes***

3.1.4) What is happening in wireshark packet numbers 1591 and 1592?
This looks like a retransmission after timeout (as there is only 1 DUP ACK after which retransmission takes place)

3.2 Now look at laptopTrace.

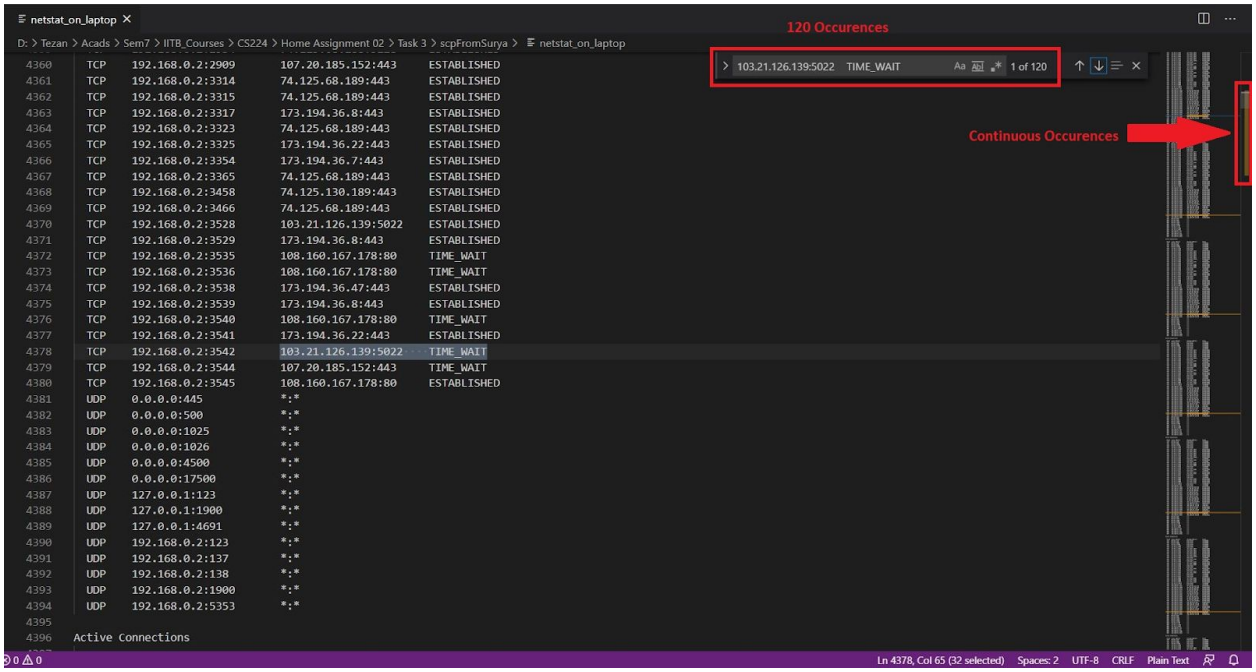
3.2.1) Which is the first wireshark packet (number) that is related to the duplicate ACK arrival at surya? What is happening in that packet?
Packet number 183 (Relative Sequence Number = 3078)
The TCP segment sent from surya to the laptop in packet 182 was not captured (indicative of loss). Hence, the packet 181 times out & a DUP ACK corresponding to packet 181 (Relative Sequence Number = 3078) is sent to surya as packet 183.

3.2.2) The TCP receive window on the laptop now has a "hole". How big is this hole (in bytes)?
ACKs have been sent upto relative sequence number = 69978
Relative Sequence Number of first packet after 'hole' = 90418 [Packet No: 182]
Thus, size of hole = ***20440 bytes***

3.3) In general look at this whole "duplicate acks" episode on both the traces and explain what happens until normal transfer begins. You must CORRELATE the two traces in your explanation. (mention fresh packet number started with)
-

3.4) Which TCP connection states do you see for the scp connection, in the netstat output?
ESTABLISHED, TIME_WAIT

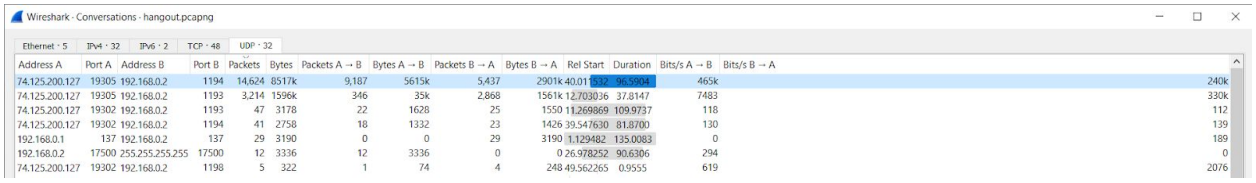
3.4.1) How long does the TIME_WAIT state last? Explain how you got this answer. (2 marks)
The TIME_WAIT state lasts for 120 seconds
Arriving at the answer:



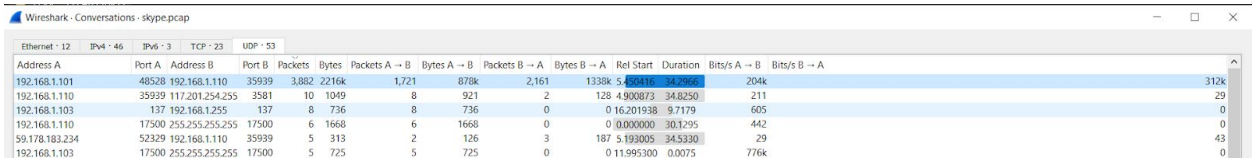
There are 120 consecutive occurrences of the TIME_WAIT state.Since the netstat command was run every second, this amounts to a total of 120 seconds.

4) The youtube, hangout and skype traces are all of multimedia traffic.

4.1) Which IP address(es) is hangout connecting to the most?
IP Address: *74.125.200.127 (seems to be a Google Server in North America)*



4.2) Which IP address(es) is skype connecting to? Who do they belong to?
IP Address: *192.168.0.101*
This address belongs to the home Wifi network



4.3) Compare and contrast these three traces, highlighting the most meaningful conclusions (do not waste time on comparisons that have no conceptual value.) Remember to mention points related to the transport layer, and any other conclusions about how the applications seem to work.
Some facts that can be used for this answer:

Skype uses the following:
- *443/TCP*
- *3478-3481/UDP*
- *50000-60000/UDP*

Hangouts uses the following:
- *TCP 19305-19309*
- *UDP 19302-19309*

Relevant conversations can be found using these ports

Youtube uses only TCP
Peculiarity: Shows a trend of 2 packets from server to client followed by 1 empty ACK packet from client to server.

Task 4:

- Details about your own network (your laptop IP configuration, your Gateway router info), your public IP address, anything about your ISP - what seems to be the address space it owns?. These questions both team members should explore.

Specifics	***Shreya***	***Tezan***
Private IP	*192.168.43.128*	*192.168.29.104*
Laptop MAC	*IntelCor_3c:e6:82 (28:b2:bd:3c:e6:82)*	*IntelCor_a6:6f:08 (40:74:e0:a6:6f:08)*
Gateway Router IP	*192.168.43.1*	*192.168.29.1*
Gateway Router MAC	*Motorola_8c:0e:1e (a8:96:75:8c:0e:1e)*	*HonHaiPr_2a:04:d6 (68:14:01:2a:04:d6)*
Public IP	*157.38.27.5*	*49.36.25.78*
ISP	*Jio*	*Jio*

- Server IPv4 and v6 addresses
  Main server for UDP: *IPv4 Address: 52.114.55.48 (IPv6 unavailable)*
  Main server for TCP:
  For Tezan: *IPv4 Address: 52.114.55.36 (IPv6 unavailable)*
  For Shreya: *IPv4 Address: 52.113.194.132 (IPv6 unavailable)*

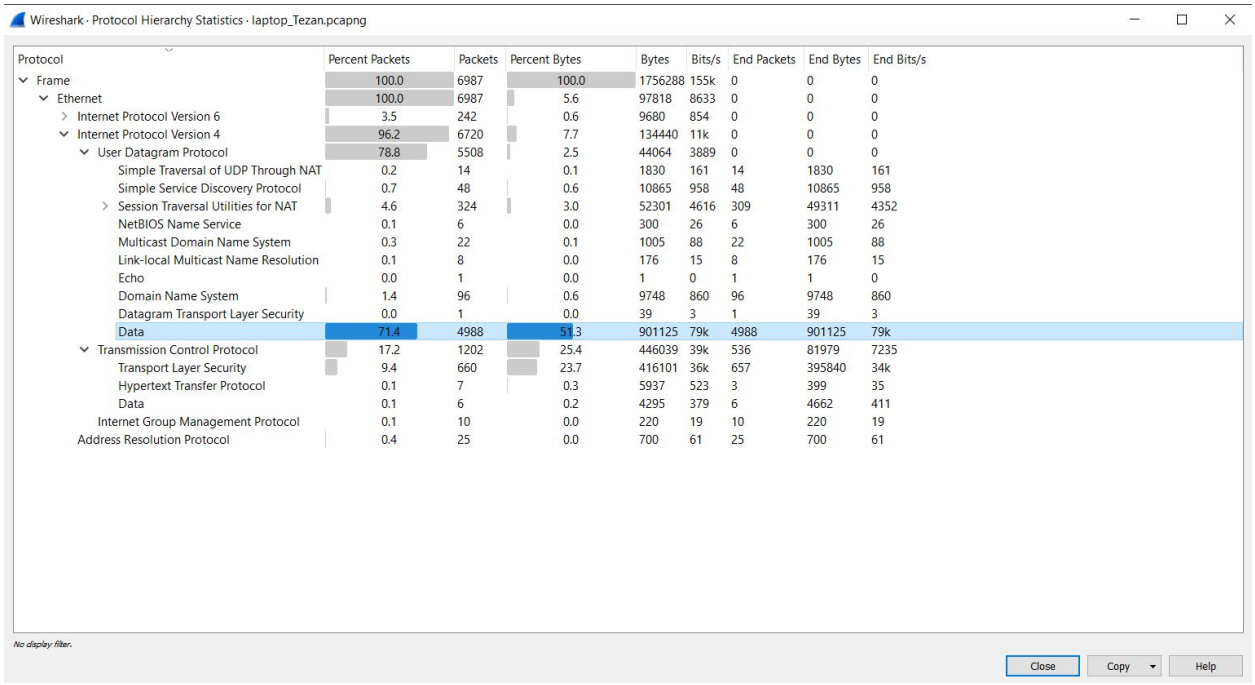
- Where are the servers located?
  Singapore (obtained from https://whatismyipaddress.com/)

- What transport layer is being used
  UDP (primarily) & TCP

- How many Simultaneous Transport layer sessions
  Considering it as total TCP+UDP connections, (maybe wrong)
  44(TCP) + 136(UDP) = 180 sessions at Tezan's end
  76(TCP) + 99(UDP) = 175 sessions at Shreya's end

- Is the communication peer to peer
  No the communication is not peer to peer. The duration of that direct connection is very less as compared to the whole conversation. The Microsoft server acts as the medium of communication (highest duration of connection and maximum data transfer)

- What's the application layer protocol for the streaming part
  No protocol identified as such from Wireshark. Simply uses ***Data*** *in* ***UDP*** *(Transport Layer) for streaming.*
  From other resources (internet), we identify that MS Teams uses ***Real-Time Messaging Protocol (RTMP)*** *for streaming.*



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	6987	100.0	1756288	155k	0	0	0
Ethernet	100.0	6987	5.6	97818	8633	0	0	0
Internet Protocol Version 6	3.5	242	0.6	9680	854	0	0	0
Internet Protocol Version 4	96.2	6720	7.7	134440	11k	0	0	0
User Datagram Protocol	78.8	5508	2.5	44064	3889	0	0	0
Simple Traversal of UDP Through NAT	0.2	14	0.1	1830	161	14	1830	161
Simple Service Discovery Protocol	0.7	48	0.6	10865	958	48	10865	958
Session Traversal Utilities for NAT	4.6	324	3.0	52301	4616	309	49311	4352
NetBIOS Name Service	0.1	6	0.0	300	26	6	300	26
Multicast Domain Name System	0.3	22	0.1	1005	88	22	1005	88
Link-local Multicast Name Resolution	0.1	8	0.0	176	15	8	176	15
Echo	0.0	1	0.0	1	0	1	1	0
Domain Name System	1.4	96	0.6	9748	860	96	9748	860
Datagram Transport Layer Security	0.0	1	0.0	39	3	1	39	3
Data	71.4	4988	51.3	901125	79k	4988	901125	79k
Transmission Control Protocol	17.2	1202	25.4	446039	39k	536	81979	7235
Transport Layer Security	9.4	660	23.7	416101	36k	657	395840	34k
Hypertext Transfer Protocol	0.1	7	0.3	5937	523	3	399	35
Data	0.1	6	0.2	4295	379	6	4662	411
Internet Group Management Protocol	0.1	10	0.0	220	19	10	220	19
Address Resolution Protocol	0.4	25	0.0	700	61	25	700	61

- What seems to be the application layer session setup protocol
  Session Traversal Utilities for NAT (STUN) *is being used for piercing the NAT at both ends & setting up the streaming session.*

- What is the bandwidth requirement per user? How does it grow with number of users (multiple teams can collaborate for this)
  - What is it with video, without video, etc
    -

- What are the round trip times, what are the one-way delays
  We define the following:
  - $RTT_{t\text{-}s}$: total time from Tezan's laptop to Shreya's laptop and back
  - $RTT_{t\text{-}server}$: total time from Tezan's laptop to Microsoft server (52.114.55.36) & back
  - $RTT_{s\text{-}server}$: total time from Shreya's laptop to Microsoft server (52.113.194.132) & back

  $RTT_{t\text{-}s}$: *Unable to calculate one-way delays or RTT (UDP packets only)*
  $RTT_{t\text{-}server}$: *0.0582 s (using iRTT mentioned in packet 118 of Tezan's trace)*
  $RTT_{s\text{-}server}$: *0.0333 s (using iRTT mentioned in packet 74 of Shreya's trace)*

- What is the route, how many hops?
  Using traceroute command at hosts, we get

Hop Count	***Sender Interface IP***	***Receiver Interface IP***
1	*192.168.43.128(Shreya's Laptop)*	*192.168.43.128 (Shreya's Gateway)*
2	*192.168.43.128 (Shreya's Gateway)*	*NA*

3	*NA*	*56.8.130.205*
4	*56.8.130.205*	*172.25.107.197*
5	*172.25.107.197*	*172.25.107.196*
6	*172.25.107.196*	*172.26.103.228*
7	*172.26.103.228*	*172.26.102.179*
8	*172.26.102.179*	*172.25.107.227*
9	*172.25.107.227*	*172.25.107.232*
10	*172.25.107.232*	*172.26.40.7*
11	*172.26.40.7*	*172.16.27.134*
12	*172.16.27.134*	*172.16.0.68*
13	*172.16.0.68*	*172.16.1.218*
14	*172.16.1.218*	*...*
...	*...*	*172.17.0.238*
N-3	*172.17.0.238*	*172.16.16.11*
N-2	*172.16.16.11*	*10.40.48.1*
N-1	*10.40.48.1*	*192.168.29.1 (Tezan's Gateway)*
N	*192.168.29.1 (Tezan's Gateway)*	*192.168.29.104 (Tezan's Laptop)*

Total Number of Hops: Certainly > 18

- What is the jitter (variance of packet interarrival times)
  -

- What is the transport layer protocol behaviour - what seems to be the nature of the packet traffic when it *leaves* your laptop - how does it change when it *arrives* at the other end? (try and see if you can plot this and compare). Packet traffic is characterized by :
  - Packet size, inter-packet delay or packet size and packet sending/arrival rate, and all of this should also be viewed in the units of bit rate
    -